MCP SERVER | NO CODE | CLOUD HOSTED

RSS / Atom Reader MCP

Pull News Feeds Directly Into Your Chat

RSS / Atom Reader MCP connects your AI client directly to any worldwide RSS or Atom feed. It acts as an instant news parser, pulling top articles and updates from external XML feeds right into your chat window for analysis.

A+ Quality Score 100/100

rss-feed | atom-feed | content-parsing | news-aggregation | xml-parsing | data-scraping

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/rss-atom-reader

SCAN TO VIEW

The connectivity layer between AI and the world's software.

Vinkius sits between AI and every application. All communication passes through Vinkius Cloud via the Model Context Protocol (MCP) — with governance, observability, and security at every layer.

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the connectivity layer where AI connects to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

> *The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*
>
> — Architecture principle

Four Pillars of the Vinkius Runtime

01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

AES-256	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	Sandboxed execution

One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

```
https://edge.vinkius.com/{token}/mcp
```

Claude · Cursor · VS Code · Windsurf · Grok · Gemini

Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken is used outside Vinkius infrastructure, the server is quarantined instantly.

Emergency Kill Switch

EU AI Act Art. 14(1) Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

Control Plane

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Incident Log

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

RSS / Atom Reader MCP

2 tools available
Cloud-hosted on Vinkius

This MCP reads and analyzes data natively from virtually any RSS or Atom feed you point it at. Instead of having to copy links, open dozens of browser tabs, and manually summarize content, this integration turns massive external news streams into summarized text inside your conversation.

It lets you monitor everything—from internal company release notes to complex market trends—without ever leaving your chat environment. You can quickly pull the latest headlines from a specific blog URL or keep an eye on a pre-configured default source for daily check-ins. With Vinkius, this MCP becomes one of the core tools in your catalog, letting you gather information flow continuously and conversationally.

Core Capabilities

01 — Check a specific news feed

You tell it a URL, and it retrieves and parses the top articles from that exact RSS or Atom feed.

02 — Read the default source

It fetches updates automatically from a standard, pre-configured feed you set up once for ongoing monitoring.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/rss-atom-reader** — connect your AI agent in three steps.

01 You first install the RSS Reader module into your AI client's MCP framework.

02 (Optional) You configure a default RSS/XML link representing a blog or journal you check regularly.

03 Then, you simply ask your agent to 'list the latest articles from our default feed,' and it reads the data.

The bottom line is that you treat external news feeds like internal chat sources, pulling structured content into plain text conversationally.

Built For

This tool is for anyone drowning in information—the market research analyst who spends hours checking dozens of industry blogs, the tech lead managing release notes, or the journalist needing real-time data streams. You're here because you hate opening ten browser tabs.

Market Research Analyst

You use it to monitor 10+ competitor news feeds simultaneously, asking your agent for a summary of the top three trends across all sources.

DevOps Engineer

You feed it internal release note URLs so your agent can pull and summarize the most critical changes from multiple service logs automatically.

Journalist/Content Creator

You track niche industry blogs or academic feeds, using the tool to extract headlines and summaries for rapid article drafting.

What Changes When You Connect

01 Stop switching tabs. Instead of navigating to multiple websites, ask your agent to read the feed directly using `read_feed`, keeping all analysis in one place.

02 Never forget a source again. Use `read_default_feed` to set up a single, reliable stream for monitoring internal updates or favorite blogs daily.

03 Turn complex XML data into plain text. This MCP handles the parsing so you just get clean, summarized articles ready for immediate use in your chat.

04 Speed is everything. You bypass paywalls and manual scraping by giving your agent direct access to public RSS/Atom streams from anywhere online.

05 Focus on insight, not navigation. The ability to pull data contextually means you spend less time gathering facts and more time analyzing them.

Real-World Applications

Monitoring competitive activity

A market analyst wants to know what the top five industry players posted in the last 24 hours. They prompt their agent: 'Using `read_feed`, fetch the latest articles from these three competitor RSS links.' The agent pulls and summarizes the findings instantly.

Tracking internal software changes

A development team needs to summarize all micro-updates across five different service repositories. They configure one feed using `read_default_feed` for the master list, letting their agent compile a single 'What's Changed' report.

Aggregating academic research

A student is researching climate science and needs to check five different university journal feeds. They ask the agent to iterate through these sources using `read_feed` to build a comprehensive background summary.

Curating content for a newsletter

A journalist wants to gather interesting articles on AI ethics for next week's issue. They tell their agent, 'Pull the top 10 from this tech blog and these three academic sources.' The tool aggregates everything into one readable block.

Patterns to Avoid

Treating it like a general search

✗ AVOID

Asking your agent to 'Find articles about X' without specifying the source. This usually leads to vague, generalized results that might not be current or structured.

✓ INSTEAD

You must tell the MCP exactly where to look. Use `read_feed` and provide a full RSS URL so the tool knows precisely which data stream to parse.

Over-complicating the default setup

✗ AVOID

Trying to manually input every single feed you might ever need into your agent's prompt. This is tedious and prone to error.

✓ INSTEAD

Set up a standard, reliable source using the `read_default_feed` tool. This way, daily check-ins require only a simple chat command.

Ignoring article limits

✗ AVOID

Asking for 'all articles' from a large feed. The agent might time out or return too much unstructured data to be useful.

✓ INSTEAD

Always specify the limit when using `read_feed`. For example, ask to retrieve the top 10 results to keep your output concise and actionable.

The Right Fit

Use this MCP if your primary need is consuming structured, highly specific content from established news or blog feeds. You should use it when you know the URL of the data source (e.g., a dedicated RSS feed). Don't use it if you are looking for general web search results, conversational answers drawn from diverse sources, or unstructured document analysis—for that, a different type of retrieval tool is better. If you need to connect content *inside* your organization, check the documentation on internal knowledge base connectors instead.

Sifting through constant updates is draining.

Today, if you want to know what's happening in a niche industry, you click. You open Chrome, navigate to the top five blogs, hit refresh on each one, and then copy-paste links or headlines into a spreadsheet. It’s manual data collection that takes half your morning.

With this MCP, you simply point your agent at those feeds. It does the heavy lifting—parsing the XML, grabbing the top stories, and delivering them summarized in plain text right where you're working. You get actionable intelligence without leaving the chat.

The RSS / Atom Reader MCP gives you structured insights.

You eliminate the need to constantly manage browser tabs, manually verify feed structures, and copy/paste messy HTML snippets. The system handles all the parsing logic for you.

What changes is that information retrieval becomes a simple conversation. You ask for updates; it delivers them cleanly.

RSS / Atom Reader: 2 Tools Available

These tools let you extract content, parse XML data structures, and scrape the latest information from various standardized feed formats.

#	TOOL	DESCRIPTION
01	`read_default_feed`	Reads the dedicated RSS feed that you set as your primary default source.
02	`read_feed`	Parses and retrieves content from any specific RSS or Atom URL you provide, letting you limit the number of results.

See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

U Determine the top 5 article posts sourced natively securely traversing strictly http://dummy.com/rss.

I securely read the feed natively using the `read_feed` call strictly passing 'http://dummy.com/rss'. Here are your 5 parsed structured news headlines:
1. Tech Launch Event Completed...
[...]

U Read the latest updates from my default feed.

I've fetched the latest 10 articles from your default feed. Here are the top headlines:
1. New Vinkius MCP Server Released
2. The Future of Conversational AI
3. Best Practices for Tool Integration
Do you want me to summarize any of these in detail?

U Fetch the single most recent article from the tech news feed.

The most recent article from the tech news feed is 'Quantum Computing Breakthrough'. It was published 2 hours ago. The summary states scientists have achieved a new milestone in qubit stability. Should I provide the full link?

Frequently Asked Questions

01 **How do I use the RSS / Atom Reader MCP if I don’t have an RSS feed?**

The tool requires a valid RSS or Atom feed URL to work. If a site doesn't offer one, you might need to check for an alternative content stream or look for a different data source connector.

02 **Can the RSS / Atom Reader MCP read from private websites?**

No, this tool is designed to pull publicly available data from open feed sources. You must point it at a public URL that publishes an accessible XML feed.

03 **What’s the difference between `read_feed` and `read_default_feed`?**

`read_default_feed` is for monitoring one reliable, long-term source. `read_feed` lets you point to any arbitrary URL when you need a quick check on a specific topic.

04 **Does the RSS / Atom Reader MCP summarize content?**

Yes, it parses the feed and delivers the articles as summarized text blocks. This makes them immediately usable for drafting or analysis in your chat session.

Go Live in 60 Seconds

Get your connection token from **cloud.vinkius.com**, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

CLIENT	WHERE TO CONFIGURE
Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"rss-atom-reader": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → Add endpoint URL
ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
Gemini	Extensions → Add MCP Server → Paste endpoint URL

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

- Ask ChatGPT
- Ask Claude
- Ask Perplexity
- Ask Gemini
- Ask Grok

READY TO CONNECT

RSS / Atom Reader is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

Start at cloud.vinkius.com →

vinkius.com · support@vinkius.com

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by RSS / Atom Reader. All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	RSS / Atom Reader MCP
Server ID	`019d7600-6cc1-726e-8c4a-98d6489e189e`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/rss-atom-reader.